

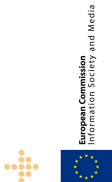


Network of Excellence on  
Engineering Secure Future Internet  
Software Services and Systems

## Network of Excellence

### D4.1 Part II:

# Engineering Secure Future Internet Services: A Research Manifesto and Agenda from the NESSoS Community



<b>Project Number</b>	:	256980
<b>Project Title</b>	:	NESSOS – Network of Excellence on Engineering Secure Future Internet Software
<b>Deliverable Type</b>	:	Other

<b>Deliverable Number</b>	:	D4.1-Part II
<b>Title of Deliverable</b>	:	D4.1 Part II: Engineering Secure Future Internet Services: A Research Manifesto and Agenda from the NESSoS Community
<b>Nature of Deliverable</b>	:	R
<b>Dissemination level</b>	:	PU
<b>Internal Document Number</b>	:	D4.1 Part II
<b>Contractual Delivery Date</b>	:	30/09/11
<b>Actual Delivery Date</b>	:	3/11/11
<b>Contributing WPs</b>	:	WP4
<b>Editor(s)</b>	:	Carmen Fernandez (UMA), Fabio Martinelli (CNR)
<b>Author(s)</b>	:	ALL
<b>Reviewer(s)</b>	:	ALL

## Abstract

In this document we advocate opportunity for establishing a discipline for engineering secure Future Internet Services, typically based on research in the areas of software engineering, of service engineering and security engineering.

The document identifies several lines of research to define a community wide research agenda in the mid-term and represents also a call for contribution to any researcher in the related sub domains in order to jointly enable the security and trustworthiness of Future Internet services.

## Keyword list

Future Internet, Software and Service Engineering, Security, Roadmap

## Document History

Version	Type of change	Author(s)
V0.1	Questionnaire evaluation and initial skeleton.	Francisco Moyano Carmen Fernández Gerardo Fernández
V0.2	New version with the structured research agenda embedded with the results of the questionnaire evaluation.	With contribution of all NESSoS partners
V0.3	Reviewed by all the partners depending on their field of expertise	ALL

## Document Review

Date	Version	Reviewer	Comment
<date of review>	<version >	<name & affiliation>	<proofed, found weaknesses, corrections ...>
11/09/2011	V0.3	ALL	

# Table of Contents

- 1 INTRODUCTION..... 6**
- 2 OVERVIEW ..... 7**
  - 2.1 Two Main Crossing Research Themes of Major Interest ..... 8
  - 2.2 Enabling Methodologies and Technologies to Enhance FI Trustworthiness ..... 9
- 3 THE SCENARIO ..... 11**
  - 3.1 Future Internet Services ..... 11
  - 3.2 Research focus on engineering secure FI services..... 12
- 4 TWO MAIN CROSSING RESEARCH THEMES OF MAJOR INTEREST ..... 13**
  - 4.1 Security Assurance..... 13
  - 4.2 Risk and Cost Aware Service Development Life Cycle (SDLC)..... 16
- 5 ENABLING METHODOLOGIES AND TECHNOLOGIES TO ENHANCE FI TRUSTWORTHINESS ..... 20**
  - 5.1 Security Requirements Engineering..... 20
  - 5.2 Secure Service Architecture and Design ..... 22
  - 5.3 Security Support in Programming Environments ..... 25
  - 5.4 Secure Service Composition and Adaptation ..... 28
  - 5.5 Run time verification and enforcement..... 29
  - 5.6 Users Security Awareness..... 31
  - 5.7 Security management..... 31
  - 5.8 Autonomic Security..... 32
- 6 CONCLUSIONS..... 34**
- REFERENCES ..... 35**
- APPENDIX A: ROADMAP QUESTIONNAIRE ..... 37**



## List of Acronyms

AOM	Aspect Oriented Modeling
CIA	Confidentiality, Integrity, Availability
FI	Future Internet
IT	Information Technology
OWASP	The Open Web Application Security Project
PKI	Public Key Infrastructure
SDLC	Service Development Life Cycle

# 1 Introduction

This document represents an initial analysis from the NESSoS project community perspective in order to define a community wide research agenda in the area of Engineering Secure Future Internet Services.

The research lines identified in this document come from the analysis and the fusion of knowledge from several sources:

- Source of knowledge has come from the NESSoS proposal and internal deliverables describing the state of the art and current research gaps in several main research strands for engineering secure services. By the time of this writing, all research areas in NESSoS had at least one state-of-the-art internal deliverable in which challenges and gaps of such areas were identified and explained.
- In order to elicit the research topics in terms of relevance, we produced a questionnaire carefully planned and organized according to Hans Schaffers (ESoCE-Net) proposal, in which four core headings were identified (Changes/Vision/Challenges/Solutions), and then it was sent via e-mail to all the NESSoS partners (see Appendix A). Once the partners had filled out the table we collected and integrated all the results. After deleting redundancies and off-topic statements, the main topics were identified. The selected topics are not of the same level of detail/granularity; nevertheless the fact that we elicited them explicitly is an indicator of their specific relevance.
- Other roadmaps in the field of the FI have been used in order to complete the information, such as the one from the Effectsplus Cluster Meeting, which also followed the same four-core headings scheme for eliciting the topics. We believe that this kind of sources is also valuable in order to keep NESSoS goals aligned with other important communities/projects/initiatives in the security area. In particular, we do plan for the future to synchronize with other communities as the ERCIM STM WG, IFIP WGs and so on for coming versions.

*The structure of the document is the following. Section 2 summarizes Section 3,4 and 5 content, in particular the main research lines we identified as promising in the future. Section 3 briefly describes the Future Internet scenario, the new threats and the specific properties that will be of foremost relevance. Section 4 describes two major research strands that represent two comprehensive and crossing themes among the ones that will follow. Section 5 describes more specific research lines worth of investigation. Section 6 concludes the document.*

## 2 Overview

This Section briefly summarizes the main findings of this report.

**Future Internet Scenario.** As the Future Internet (FI) arises, long-standing issues as well as new problems appear that need to be addressed in the area of engineering secure software-based services. Just to mention, we are witnessing the emergence of new and unprecedented models for service-oriented computing for the Future Internet: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Moreover, it is of paramount importance how to securely manage the huge growth of data and information across organizational domains. As a consequence, services are being outsourced by companies and moved into clouds. These models have the potential to better adhere to an economy of scale and have already shown their commercial value fostered by key players in the field. Nevertheless, those new models present change of control on the applications that will run on an infrastructure not under the direct control of the business service provider. For business critical applications this could be difficult to be accepted, when not appropriately managed and secured.

In the following we will sketch the main topics of research or properties that we identified as crucial in the area of Secure Service Engineering (see also Figure 1).

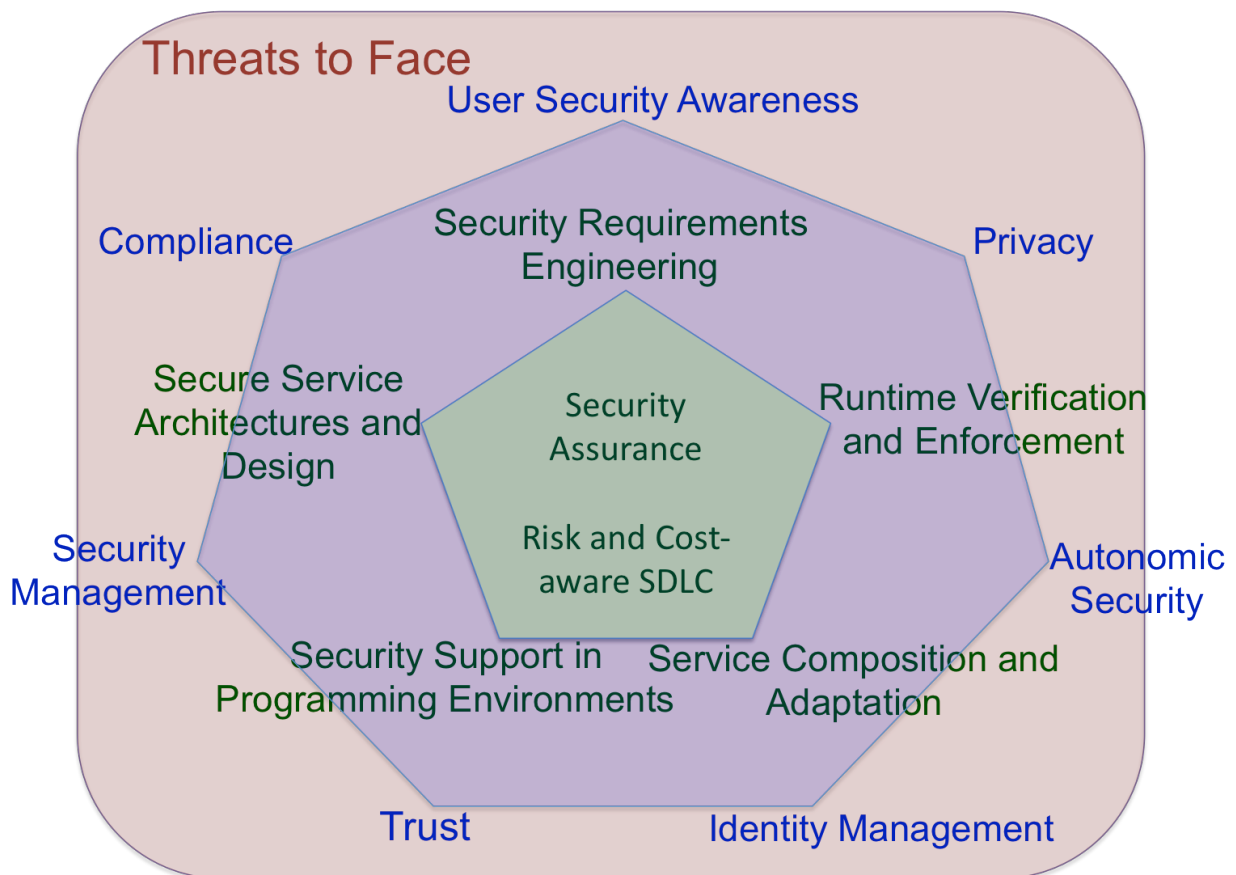


Figure 1 Main Topics and Research Areas

**Threats to face.** In the Future Internet scenario, a new massive growth of cyber-crimes/cyber-attacks is expected. Future Internet will boost malware propagation through different kind of networks, either by moving physically (e.g. USB sticks, car-car, mobile devices) or by looking for communication gateways in the surroundings. This jump in communication capabilities along with the availability of more computing power, can be used by attackers to compromise resource constrained devices connected to the Internet, in order to access potentially dangerous resources and information sources. The implicit and explicit interdependences for offering IT services (often managed by different stakeholders) make it difficult to predict system behaviour as well as to enforce properly security policy and more general compliance to cross-border legislations. *Prosumers* (producers&consumers) creating and using services will lose control on their data (including the one regarding their personal sphere). Cyber-attacks are increasing in the scope and impact, targeting critical information infrastructures. There is also an increasing relationship between security and safety issues due to the proliferation of embedded system in our everyday life.

**Properties to be ensured in FI.** Several properties should be ensured for Future Internet: *Compliance, Privacy, Trust, and Identity protection*. These topics are currently becoming more and more important due to the increase of interoperability and heterogeneity concerns within the FI. It is worth to mention that location-privacy will be crucial in FI scenarios. Also, the “right to be forgotten” is challenging. New schemas for identity management are required, since identity is emerging as an additional Internet layer with high impact within the FI. Also, trust management among boundaries will become of paramount importance in order to gain a tight integration of things, services and humans for an efficient and useful FI.

The need to embed security issues in the Future Internet service development life-cycle becomes evident. The next two sub-sections highlight the main research strands we consider worth of investigation in the mid-term.

## **2.1 Two Main Crossing Research Themes of Major Interest**

*Security Assurance during SDLC:* Assurance will play a central role in the development of software-based services to provide confidence about the desired security level. Assurance will be treated in a holistic manner as an integral constituent of the development process (design for assurance), seamlessly informing and giving feedback at each stage of the software life cycle by checking that the related models and artefacts satisfy their functional and security requirements and constraints. Since the choice of appropriate assurance methods depends on several factors including the concrete application context and the desired level of assurance, this activity will cover a correspondingly broad range of assurance methods that jointly offer full development cycle support. Quantitative notions of security (including metrics) will allow

to have systems able to trade-off among several requirements in a rationale way and considering multiple factors (including energy consumption for the protection mechanisms).

***Risk and Cost-aware SDLC:*** It is hard to master risks and costs in service infrastructures, especially with the uprising of clouds and complex infrastructures. Some relevant goals are achieving traceability between risk and development models, managing evolving risks, and assessing risks at run-time. The definition and refinement of more precise economic security measures is necessary as well. As a matter of fact, the value of security solutions and their return on investment must be clearly demonstrated from a business oriented perspective. The value of the chosen security solutions has to be derived from the risk analysis. The net value of the investment must be derived by analysing the cost that comes with creating security solutions and implementing security measures. The integration of risk and cost analysis in the whole SDLC, and an extension of the overall approach towards execution time, are the necessary response to these needs.

## **2.2 Enabling Methodologies and Technologies to Enhance FI Trustworthiness**

***Security Requirements Engineering:*** The highly interconnected environment of the Future Internet, which mixes various infrastructure resources with application functionalities, inherits security risks from the classical Internet at the same time that it creates new and more complex security requirements challenges, such as conflicts resolution amongst stakeholders interests or new types of security and privacy requirements, such as location-privacy. Other issues to address have to do with evolving requirements through the SDLC, having business-socio-economic requirements into account, and validating the completeness of the requirements.

***Secure Service Architectures and Design:*** We need to increase the capability of designing secure software-based service systems for the internet of the future, to analyze security and ensure compliance of the underpinning architectures, and to include identification, assessment and improvement of design principles in order to enhance those architectures in terms of flexibility, modularity and composability, what will facilitate the integration of novel security services as the Future Internet scenarios evolve. Several approaches for modeling and developing architectures for SoA must be covered, including architecture design languages extended with security features as well as design approaches as design by contract (and its ‘relative’ security by contract) and correct-by-construction (stepwise refinement). Other areas of interest that will be investigated for Future Internet paradigms (such as SaaS) include Model-driven architecture, Model-driven security, and other Model-driven development approaches.

***Security Support in Programming Environments:*** This research area covers new programming platforms that deliver development and runtime environments for trustworthy application code to be executed in the complex application scenarios depicted by the Future

Internet. We will also address language based security, as well as secure coding principles and practices. Research will both be based on language design and implementations, including middleware and run-time environment. Type systems, verifying compilers, support for run-time property verification and enforcement will be addressed here as well. Programming principles and constructs will be investigated in order to ease secure service development and composition for the new application scenarios. Code signatures as well as code instrumentations, aspect oriented and other composition techniques for security and secure execution environments are also in the scope of this area.

***Service Composition and Adaptation:*** The integration and interoperability of services in order to tailor and enhance new services require adapting the service interfaces at different levels, including the semantic level. Other aspects to consider include assessing the trustworthiness of composition of services as well as and composing security measures.

***Run-time Verification and Enforcement:*** Several levels of protection must be available. Thus run time enforcement of security properties is useful and must be further investigated. In this ambit, one of the most important gaps to overcome arises from the fact that security policies are typically formulated at high levels of abstraction whereas the monitors observe system events that are low-level. Model-driven approaches would be useful also in this context.

***Users Security Awareness:*** users demand increasing the knowledge on the risks they are exposed to when using a service or a system. Changes in the context under which a service is running should be clearly informed to the user. This can be achieved by relying on good security usability mechanisms for end-users, and on friendly privacy controls.

***Security Management:*** It is clear that secure services –especially the security features and related subsystems - should be supported with appropriate management support in order to observe the “quality of protection” in production systems at run time, and in order to implement the necessary measures for dealing with new threats and attacks, and possibly also with security incidents that require modification of the service implementation and/or its deployment environment.

***Autonomic Security:*** FI services need to execute on a context, which is set up based on the service environment information. This would allow “autonomically” choosing different security mechanisms depending on certain levels of security required for a specific context. It also allows changing this context through the service execution life. For these purposes, new reasoners (decision making processes) need to be developed in order to intelligently exploit the service environment information, and thus, to predict security reconfiguration needs according to changes in this environment.

# 3 The Scenario

## 3.1 Future Internet Services

### The Future Internet Paradigm

The concept named Future Internet (FI) aggregates many facets of technology and its practical use, often illustrated by a set of usage scenarios and typical applications. The Future Internet may evolve to use new infrastructures, network technologies and protocols in support of a growing scale and a converging world, especially in light of smaller, portable, ubiquitous and pervasive devices. Besides such a network-level evolution, the Future Internet will manifest itself to the broad mass of end users through a new generation of services (e.g. a hybrid aggregation of content and functionality), service factories (e.g., personal and enterprise mash-ups), and service warehouses (e.g., platform as a service). One specific service instance may thus be created by multiple service development organizations, it may be hosted and deployed by multiple providers, and may be operated and used by a virtual consortium of business stakeholders. While the creative space of services composition is in principle unlimited, so is the fragmentation of ownership of both services and content, as well as the complexity of implicit and explicit relations among participants in each business value chain that is generated. In addition, the user community of such FI services evolves and widens rapidly, including masses of typical end users in the role of *prosumers* (producing and consuming services). This phenomenon increases the scale, the heterogeneity and the performance challenges that come with FI service systems.

### Threats and Risks

Yet, this new paradigm presents new vulnerabilities and risks as the number of trust domains in an application will be multiplied, the size of attack surfaces grows and so does the number of threats. Furthermore, the Future Internet will be an intrinsically dynamic and evolving paradigm where, for instance, end users are more and more empowered and therefore decide (often on the spot) on how content and services are shared and composed. This adds an extra level of complexity, as both risks and assumptions are hard to anticipate. Moreover, both risks and assumptions may evolve; thus they must be monitored and reassessed continuously.

Indeed, there is also a growth of successful attacks on ICT-service systems, both in terms of impact and variety. This obviously harms the economic impact of Future Internet services and causes significant monetary losses in recovering from those attacks. In addition, this induces users at several levels to lose confidence in the adoption of ICT-services

### Properties

In addition to the usual confidentiality, integrity and availability properties, the evolution of the Future Internet obviously puts the focus on the **trustworthiness** of services. Multiparty service systems are not entirely new, yet the Future Internet stretches the present know how on building secure software services and systems: more stakeholders with different trust levels are

involved in a typical service composition and a variety of potentially harmful content sources are leveraged to provide value to the end user. This is attractive in terms of degrees of freedom in the creation of service offerings and businesses. New Internet services will have to be provided in the near future, and security breaches in these services may lead to large financial loss and damaged reputation. Compliance to national and international legislations/regulations/policies is a main aspect to be ensured. Also privacy will have a significant role in the FI scenarios. The pervasiveness of FI services one hand will allow an unprecedented gathering of information, as well as, fostering new threats to personal identities and habits.

### 3.2 Research focus on engineering secure FI services

Our focus is on the creation and correct execution of a set of methodologies, processes and tools for secure software-services development. This typically covers requirements engineering, architecture creation, design and implementation techniques. However this is not enough! We need to enable **assurance**: approving that the developed software service is secure. Assurance must be based on **justifiable evidence**, and the whole process **designed for assurance**. This would allow the uptake of new ICT-services according to the latest Future Internet paradigms, where services are composed by simpler services (provided by separate administrative domains) integrated using third parties infrastructures and platforms. The need of managing the intrinsic **modularity and compose-ability** of these architectures, traditionally shared with commercial off the shelf components (COTS), should drive the development of corresponding methodologies. User requirements may change and evolve during time thus demanding mechanisms for **adaptation and evolution** of the services/systems. User empowerment with mechanisms for **policy** management would be useful.

Clearly industry needs to prioritize its efforts in order to improve its return on investments (ROI). Thus, embedding **risk/cost analysis** in the SDLC is currently one of the key research directions in order to link security concerns with business needs and thus supporting a business case for security matters.

The NESSoS community addresses the early phases of the development process of services, bearing in mind that the discovery and remediation of vulnerabilities during the early development stages saves resources (prevention is better than the cure).

In the sequel of this document we elaborate on the relevant sub domains and techniques that we consider useful for engineering secure Future Internet services in the future.

## 4 Two Main Crossing Research Themes of Major Interest

Engineering secure Future Internet services demands for two traversal issues, security assurance and risk and cost management during SDLC.

### 4.1 Security Assurance

The main objective is to enable assurance in the development of software based services to ensure confidence about their security level. Our core goal is to incept a transverse methodology that enables to manage assurance throughout the service and software development life cycle (SDLC). The methodology is based on several strands: A first sub-domain covers early assurance at the level of requirements, architecture and design. A second sub-domain includes the more conventional and complementary assurance techniques based on implementation. Additionally, we consider also quantitative notions of security (e.g. security metrics).

#### 4.1.1 Assurance during the Early Stages of SDLC

Early detection of security failures in Future Internet applications reduces development costs and improves assurance in the final system. This first strand aims at developing and applying assurance methods and techniques for early security verification. These methods are applied to abstract models that are developed from requirements to detailed designs.

One main area of research is step-wise refinement of security, by developing refinement strategies, from policies down to mechanisms, for complex protocols, services, and systems. This involves the definition of suitable service and component abstractions (e.g., secure channels) and the setup of the corresponding reasoning infrastructure (e.g., facts about such channels). Moreover, we will extend our refinement framework with *compositional* techniques for model-based secure service development. Model decomposition supports a divide-and-conquer approach, where functional and security-related design aspects can be refined independently. Model composition must preserve the refinement relation and component properties. Our aim is to offer developers support for smoothly integrating security aspects into the system development process at any step of the development.

There is an increasing demand of models and techniques to allow the formal analysis of secure services. The objective is to develop methodologies, based on formal mappings from constraint languages, to other formalisms for which theorem proving and/or (semi-) decision procedures are available, to support formal (and, when possible, automated) reasoning about the security policies models.

The methodologies must be supported by automatic protocol verification tools for the verification of Future Internet protocols. The planned extensions require not only significant efficiency improvements, but also the ability to deal with more complex primitives and security properties. Moreover, the Dolev-Yao attacker model used by these tools needs to be extended to include new attack possibilities such as adaptive corruptions, XSS attacks, XML injection, and guessing attacks on weak passwords. In addition, in order to increase the degree of assurance, there is the need to extend model checking methods to enable automatic generation of protocol correctness proofs that can be independently verified by automated theorem proving.

One needs however, to bear in mind that model abstractions that allow performing automatically security model checking are several times too coarse. This means that formally proving that a security specification provides the desired security services is still challenging.

Furthermore, metrics for the most pressing security problems in FI, such as privacy of users or isolation in cloud computing, especially under metrology basis, are still lacking. Moreover, there is a need for more formal metrics, since most of metrics are only informally described. Lastly, validation and comparison frameworks for security metrics are required to objectively evaluate their quality under actual, huge datasets.

Assurance will play a major role in service-oriented applications, leading to a design for assurance methodology that tries to make as easy as possible assurance cases definitions through the whole SDLC, as well as connecting assurance cases amongst different phases of SDLC. In addition, and given the rising of new types of business processes based on services outsourcing, it is foreseen the existence of frameworks to analyse the compliance of a system process when it involves the outsourcing of services. In this sense, developing adequate certification and audit frameworks pose a very significant challenge.

#### **4.1.2 Security Assurance in Implementation**

Several complementary assurance techniques are available to ensure the security at the level of an implementation. Security policies can be implemented correctly by construction through a rigorous secure programming discipline. Internet applications can be validated through testing. It is possible to develop test data generation that specifically targets the integration of services, access control policies or specific attacks. Moreover, implementations can be monitored at run-time to ensure that they satisfy the required security properties.

Several activities are related to **secure programming**. This strand addresses a comprehensive solution for program verification, with a particular focus on session management in concurrent and distributed service compositions.

- *Programming for verifiable security*: The Future Internet will reinforce the prominence of highly distributed and concurrent applications, increasing the need for methodologies that prevent security holes that exploit the computational infrastructure. The objective is

to develop a discipline of secure programming based on verifiable security, using program analyses and verification methods. We need to develop enforcement mechanisms that combine different verification methods and allow enforcing a wide range of policies (of information flow and resource usage).

- *Support for secured session management*: A specific issue for web service security is the proper processing of the message stack. For instance, a BPEL process may have several instances or sessions running concurrently and among these instances several might be accepting some incoming message at the same time. Hence for the run-time reliability of the system it is crucial to assign the right message to the right session. This dispatching operation needs to be carefully designed from security requirements.

In addition, an important set of research activities concerns **testing**, which complements programming and coding. We can consider three aspects, that although not comprehensive, characteristic for service-oriented applications in the Future Internet: penetration testing that leverages on the high-level models that are generated in early stages of the software life cycle, automated generation in XML-based input data to maximize the efficiency in the security testing process, and testing of policies that are the typical high-level front end of a complex service composition. The latter part will focus on access control policies.

- *Model-based penetration testing*. Penetration testing consists in evaluating the robustness of a (current or Future) Internet application to well-known attacks. We need to investigate models to guide a penetration testing tool for interactive testing, which should be automated as much as possible. Given an implementation, and a model for a common type of attack or risk, we want to do penetration testing to see if this implementation suffers from the vulnerability. We assume the models are transition systems. These inputs are fed into a penetration testing tool to guide the user in interactive model-driven penetration testing.
- *Automatic generation of test data for web applications*. The input domain of a web application can be modelled by an XML schema. In that case a test data is an XML file conforming to the schema. The major issue for test generation is that there can be an infinite number of XML files that conform to the schema. We need to explore several approaches for automatic test data generation. A first approach is based on the partitioning of the input domain and then automatically generate instances in each range of the partition. A second approach consists of sampling the input domain through combinatorial testing techniques.

### **Towards a traverse methodology**

Security concerns are specified at the business-level but have to be implemented in complex distributed and adaptable systems of FI services. We need comprehensive assurance techniques in order to guarantee that security concerns are correctly taken into account through the whole SDLC. A chain of techniques and tools crossing the above areas is mandatory.

### **4.1.3 Security Metrics**

Measurements are essential for the objective analysis of system security. Metrics can be used directly for computing risks (e.g., probability of threat occurrence) or indirectly (e.g., time between antivirus updates). Security metrics in Future Internet applications become increasingly important (although still very challenging). Service-oriented architectures demand for assurance indicators that can explicitly measure the quality of protection of a service, and hence indicate the effective level of trustworthiness. These metrics should be assessable by and communicable to third parties. Clients want to be sure that their data is well protected when it is outsourced to other domains over which the client has only limited or no control. We need to define formal metrics and measurements that can be practically calculated. Compositional calculation approaches will be studied in this context. Many of the proposed metrics will be linked to and determined by the various techniques in the Engineering process.

### **4.1.4 Threats**

We envisage several threats towards the development and provisioning of secure FI services should the scientific community at large fail to pursue the above described agenda.

The increasing complexity of services demands not only the ability to build secure systems, but also to be able to prove and certify service security levels to third parties than will then utilize these services for their own purposes (new services). This is a complex task, requiring a chain of proofs and assurances cases and techniques during the whole SDCL. Indeed, it is important to stress that the assurance techniques associated with the different phases of the SDLC such as design verification, testing and debugging, and runtime verification and enforcement are complementary yet all necessary. This means that each of these techniques independently contributes towards improving security assurance. Therefore, we have to invest further research efforts in order to improve all of these techniques and make them fit to face the challenges posed by the Future Internet. More generally, software with vulnerabilities presents a threat not only to European competitiveness but also to European economy and critical infrastructures. We will witness situations where the vendor of software might not provide a patch in time to minimize the risks of attacks, which take advantage of the given vulnerability. The issues such as security assurance (together other phases of secure service engineering) also address the crucial need to evaluate software coming from unknown or not well-known European sources.

## **4.2 Risk and Cost Aware Service Development Life Cycle (SDLC)**

Risk management is coordinated activities to direct and control an organization with regard to risk, and involves risk assessment and the identification of treatment options for unacceptable risks [ISO09]. Our research agenda for a risk and cost aware SDLC highlights challenges that are FI characteristic and for which traditional methods for risk identification, risk modelling and risk analysis provide little specialized support. Generally, there is the need for a

methodology to support a risk and cost aware SDLC for secure FI services. Such a life cycle model aims to ensure the stakeholders' return of investment when implementing security measures during various stages of the SDLC. We can envision several aspects of this kind of SDLC:

- *Process*: The methodology for risk and cost aware SDLC should be based on an *incremental and iterative process* that is accommodated to an incremental software development process. While the software development proceeds through incremental phases, the risk and cost analysis will undergo new iterations for each phase. As such the results of the initial risk and cost analyses will propagate through the software development phases and become more refined. In order to support the propagation of analysis results through the phases of the SDLC one needs to develop methods and techniques for the refinement of risk analysis documentation. Such refinement can be obtained both by refining the risk models, e.g. by detailing the description of relevant threats and vulnerabilities, and by accordingly refining the system and service models.
- *Aggregation*: In order to accommodate to a modular software development process, as well as effectively handling the heterogeneous and compositional nature of Future Internet services that also involves the perspective and requirements of several competing stakeholders, one needs to focus on a modular approach to the analysis of risks and costs. In a compositional setting, also risks become compositional and should be analysed and understood as such. This requires, however, methods for aggregating the global risk level through risk composition, which will be investigated.
- *Evolution*: The setting of dynamic and evolving systems furthermore implies that risk models and sets of chosen mitigations are dynamic and evolving. Thus, in order to maintain risk and cost awareness, there is a need to continuously reassess risks and identify cost-efficient means for risk mitigation as a response to service or component substitution, evolving environments, evolving security requirements, new stakeholders emerging, etc., both during system development and operation. Based on the modular approach to risk and cost analysis one needs methods to manage the dynamics of risks. In particular, the process for risk and cost analysis is highly iterative by supporting updates of global analysis results through the analysis of only the relevant parts of the system as a response to local changes and evolvments.
- *Interaction*: The methodology of this strand spans the orthogonal activities of security requirement engineering, secure architecture and design, secure programming as well as assurance and the relation to each of these ingredients must be investigated. During security requirements engineering risk analysis facilitates the identification of relevant requirements. Furthermore, methods for risk and cost analysis offer support for the prioritisation and selection among requirements through e.g. the evaluation of trade-off between alternatives or the impact of priority changes on the overall level of risks and cost. In the identification of security mechanisms intended to fulfil the security

requirements, risk and cost analysis can be utilised in selecting the most cost efficient mechanisms. The following architecture and design phase incorporates the security requirements into the system design. The risk and cost models resulting from the previous development phase can at this point be refined and elaborated to support the management of risks and costs in the design decisions. Moreover, applying cost metrics to design models and architecture descriptions allows early validation of cost estimates. Such cost metrics may also be used in combination with security metrics for the optimisation of the balance between risk and cost. The assurance techniques can therefore be utilised in providing input to risk and cost analysis, and in supporting the identification of means for risk mitigation based on security metrics.

#### **4.2.1 Other Challenges**

A formal foundation for risk management may serve as a basis for rigorous analysis and reasoning about risk by means of formal methods. The first challenge in risk analysis and management from a formal methods perspective is to find a formal semantic foundation of risk that is sufficiently general and expressive for the task. The second one, and possibly a harder challenge, is to utilize the formal foundation, together with the principles of formal methods, to define methods to support risk analysis and surrounding activities.

Risk methodology validation and integration are crucial issues as well. The former is required to analyse and compare objectively the strengths and weaknesses of different risk methodologies. The latter allows combining different risk frameworks in order to leverage the strengths of the frameworks and suppressing their limitations.

Some efforts are already being done on achieving higher level of abstractions for security SW and risk and cost-awareness, on dynamic risk monitoring and on common semantics for development and risk analysis models, in order to achieve traceability amongst both models. Nevertheless, other solutions are necessary. Run-time risk assessment is required since risk changes very frequently and static risk analysis performed once or twice a year is not enough. Run-time re-configurability of security based on risk management requires further research. Other areas include achieving modularity, identifying common cause of failures, analysing inconsistencies and handling changes in risk analysis documentation, evolving risks, allowing composition and decomposition of risk models and dynamic contracts-based risk sharing via certification.

#### **4.2.2 Threats**

We envisage several threats towards the development and provisioning of secure FI services should the scientific community at large fail to pursue the above described agenda. In particular:

- Industry and service providers need to ensure properties such as compliance, privacy, trust and identity protection while making business. Without methods to ensure return on investment in security, security may fail the competition with other business priorities. The ROI in security during the SDLC must therefore not only be ensured, but also clearly demonstrated at a business level.
- Risk and cost management in the FI setting must be supported by methods, techniques and tools to handle the highly dynamic, compositional and heterogeneous nature of FI. Traditional risk analysis methods are largely monolithic in the sense that systems are understood and analysed as a whole. Without a modular approach to risk analysis, a full analysis may have to be conducted anew whenever services or systems are recomposed. There is hence a risk that traditional methods become too heavy and costly for the FI setting, and that risk models and risk analysis results quickly become invalid and outdated.
- Related to the latter is the lack of methods and techniques to handle change and evolution. When systems and services change and evolve, so do risks and should be modelled and analysed as such. With traditional methods, previous analysis results may become outdated and risk analysis efforts can be in vain. Moreover, lack of appropriate means for run-time risk assessment of the frequently changing FI services may be highly insufficient for continuously keeping the security risk picture up to date.

# 5 Enabling Methodologies and Technologies to Enhance FI Trustworthiness

## 5.1 Security Requirements Engineering

The main focus of this research strand is to enable the modelling of high-level requirements that can be expressed in terms of high-level concepts such as compliance, privacy, trust, and so on. These can be subsequently mapped into more specific requirements that refer to devices and to specific services. A key challenge is to support dealing with an unprecedented multitude of autonomous stakeholders and devices – probably one of the most distinguishing characteristics of the FI.

The need for assurance in the Future Internet demands a set of novel engineering methodologies to guarantee secure system behaviour and provide credible evidence that the identified security requirements have been met from the point of view of all stakeholders. The security requirements of Future Internet applications will differ considerably from those of traditional applications. The reason is that Future Internet applications will not only be distributed geographically, as are traditional applications, but they will also involve multiple autonomous stakeholders, and may involve an array of physical devices such as smart cards, phones, RFID sensors and so on that are perpetually connected and transmit a variety of information including identity, bank accounts, location, and so on. Some of these transactions might even happen transparently to the user; for example, a person's identity could be seamlessly communicated by a personal device to the store she is entering to do the shopping. Addressing concerns about identity theft, unauthorized credit card usage, unauthorized transmission of information by third-party devices, trust, privacy, and so on are critical to the successful adoption of FI applications.

Service-orientation and the fragmentation of services (both key characteristics of FI applications) imply that a multitude of stakeholders will be involved in a service composition and each one will have its own security requirements. Hence, eliciting, reconciling, and modelling all the stakeholders' security requirements become a major challenge [BGG+04]. Multilateral Security Requirements Analysis techniques have been advocated in the state of the art [GBS06] but substantial research is still needed. In this respect, agent-oriented and goal-oriented approaches such as Secure Tropos [GMZ06] and KAOS [DLF93] are currently well recognized as means to explicitly take the stakeholders' perspective into account. These approaches will represent a promising starting point but need to be uplifted in order to be able to cope with the level of complexity put forward by FI applications. New requirements frameworks and languages that take legislative constraints, as well as socio-technical and economic aspects into account, are needed in order to manage data through multiple domains. Indeed, it is important that security requirements are addressed from a higher-level perspective, e.g., in terms of the actors' relationships with each other. Unfortunately, most current requirements engineering approaches consider security only at the technological level. In other

words, current approaches provide modelling and reasoning support for encryption, authentication, access control, non-repudiation and similar requirements. However, they fail to capture the high-level requirements of trust, privacy, compliance, and so on. It is also essential to analyse how security may impact on other functional and non-functional requirements, including Quality of Service/Protection (QoS/P), both at design-time and at run-time.

This picture is further complicated due to the vast number and the geographical spread of smart devices stakeholders would deploy to meet their requirements. Sensor networks, RFID tags, smart appliances that communicate not only with the user but with their manufacturers, are examples of such devices. Such deployments inherit security risks from the classical Internet and, at the same time, create new and more complex security challenges. Examples include illicit tracking of RFID tags (privacy violation) and cloning of data on RFID tags (identity theft). Applications that involve such deployments typically cross organization boundaries.

In light of the challenges and principles highlighted above, we identify the following detailed research objectives:

- The definition of techniques for the identification of all stakeholders (including attackers), the elicitation of high-level security goals for all stakeholders, and the identification and resolution of conflicts among different stakeholder security goals;
- The refinement of security goals into more detailed security requirements for specific services and devices;
- The identification and resolution of conflicts between security requirements and other requirements (functional and other quality requirements);
- The transformation of a consolidated set of security requirements into security specifications.

The four objectives listed above obviously remain generic by nature, one should bear in mind though that the forthcoming techniques and results will be applied to a versatile set of services, devices and stakeholder concerns.

### **5.1.1 Threats**

There are several threats in the case the previous research strand is not properly addressed. In particular, business risks are increasingly related to operational risk and eventually to IT security risks. The lack of the capability to properly embed specific FI security requirements in the SDLC and the lack of analysis capability when considering conflicting multi-stakeholder interests will dramatically limit the effectiveness and impact of new FI services.

## 5.2 Secure Service Architecture and Design

FI applications entail scenarios in which there exist a huge amount of heterogeneous users and a high level of composition and adaptation is required. These factors increase the complexity of applications and make it necessary to leverage existing mechanisms and methodologies for software construction as well as researching about new ways to take this complexity into account in a holistic manner. These applications enable pervasive, ubiquitous scenarios where multiple users, devices, third-party components interact continuously and seamlessly, so security enforcement mechanisms are indispensable. The design phase of the software service and/or system is a timely moment to enforce and reason about these security mechanisms, since by that phase one must have already grasped a thorough understanding of the application domain and of the requirements to be fulfilled. Furthermore, at design-time a preliminary version of the application architecture has been produced.

The software architecture encompasses the more relevant elements of the application, providing either a static or/and a dynamic view of the application. A more comprehensive definition can be found in [BCK04], where it is defined as “the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them”. In the context of security, externally visible properties would include proof of identity, enforcement of access control by a user of the component, privacy, a line of defence against denial of service, and so on. The security architecture for the system must enforce the visible security properties of components and the relationships between them. All this information makes it feasible to enforce, assess and reason about security mechanisms at an early phase in the software development cycle.

The research topics one must focus on in this subarea relate to model-driven architecture and security, the compositionality of design models and the study of security design patterns for FI services and applications.

As for the first element the aim is to support methodologies that enable separation of concerns in order to reason about each concern (*e.g.*, security, functionality, GUI, etc.) in isolation, in domain specific terms and with the most adapted techniques. According to the model-driven approach, these models can then be composed to form a global design, in such a way that they all preserve certain properties. So, it would be possible to specify a first high-level model with some high-level security policies. Then, by automation, this model could be converted into another more specific model, in which the security policies become more detailed, closer to the enforcement mechanisms that will fulfil them. This process should be applied until a basic version of the application architecture can be released.

The integration of security aspects into this paradigm is the so-called model-driven security [CSB+08], leading to a design for assurance methodology in which every step of the design process is performed taking security as a primary goal. A way of carrying out this integration includes first decomposing security concerns, so that the application architecture and its

security architecture is decoupled. This makes possible for architects to assess more easily trade-offs among different security mechanisms, simulate security policies and test security protocols before the implementation phase, where changes are typically far more expensive.

In order to achieve this, it is first needed to convert the security requirements models into a security architecture by means of automatic model transformations and manual refinements. These transformations represent a crucial step in the development since they bridge the gap between the requirements that belong to the problem-domain and the architecture and design models that belong to the solution-domain. In the context of security modelling, it is extremely relevant to invent ways to model the usage control property, which encompasses traditional access control, trust management and digital rights management and goes beyond these building blocks in terms of definition and scope. Transformation patterns, capture systematic engineering principles that can be reused in various FI scenarios, to map the high-level policies established at requirements stage into low-level, enforceable policies at run-time. Furthermore note that FI scenarios include Cloud and GRID services and although some work has already been made in the area [RFL10], further research is necessary to find out what kind of security architecture is required in the context and how to carry out the decomposition of such fairly novel architectures.

In order to grasp a comprehensive understanding of the application as a whole, it is required to integrate all the different models that capture various views on the system into a unified one. This process is called composition [WMA+07, FFR+07] and, as a recent work suggests [MFB+08], it is possible to perform it at run-time, adding a new level of flexibility and adaptation for FI applications. Regarding composition, several topics will be studied. First, it is desirable to define contracts for model composition, in such a way that only correct compositions are allowed, limiting the propagation of design flaws through the models. Second, a major challenge will emerge from the large amount of heterogeneity that will be found in FI: this will mean composing models that have heterogeneous execution semantics and understanding the relations and potential conflicts between these. Finally, adaptation of composite services is a key area of interest. FI scenarios are very dynamic, so threats in the environment may change along the time and some reconfiguration may be required to adapt to that changes.

Another research focus is on design patterns and on reusable architectural know-how. A design pattern is a general repeatable solution to a commonly occurring problem in software design. Design patterns, once identified, allow reuse of design solutions that have proved to be effective in the past, reducing costs and risks usually arisen by uncertainty, leveraging a risk and cost-aware. There are large catalogues and surveys on security patterns available [YWM08, OG], but the FI applications yet to come and the new scenarios enabled by FI need to extend and tailor these catalogues. In this context, the first step is studying the patterns currently available and, what is more important, to analyse the relationships amongst them [KWF08], identifying those which may be useful for FI scenarios.

### 5.2.1 Threats

We distinguish two critical areas for the design and architecture of secure services:

- **Early design.** If we do not set effective methods and tools to deal with security concerns in design models the major threat we will face will be the increasing cost to deploy, fix and maintain security mechanisms. Indeed, security concerns are tightly related to other concerns such as functionality or human interfaces. If we can design abstract models for these concerns, then, the cost to analyze interactions and fix the models until reaching a satisfactory trade-off between the concerns, can be kept reasonable. On the other hand, if these analysis cannot be performed on abstract models, they are much more difficult to understand at the code level, and even more difficult to fix and maintain, because of all the technical details that must be introduced in the implementation.
- **Model composition.** Because of the highly dynamic nature of FI applications, reconfiguration and adaptation are essential mechanisms for the success of FI. Thus, we need rigorous and systematic techniques for composing models to analyse interactions before reconfiguration. The major threat if we cannot achieve safe and effective model composition will be limited abilities for adaptation and thus increased risks of failures in front of major changes in the environment.

### 5.3 Security Support in Programming Environments

Security Support in Programming Environments is not new; still it remains a grand challenge, especially in the context of FI services. Securing Future Internet Service is inherently a matter of secure software and systems. The context of the future internet services sets the scene in the sense that (1) specific service architectures will be used, that (2) new types of environments will be exploited, ranging from small embedded devices (“things”) to service infrastructures and platform in the cloud, and (3) a broad range of programming technologies will be used to develop the actual software and systems.

The search for security support in programming environments has to take this context in account. The requirements and architectural blueprints that will be produced in earlier stages of the software engineering process cannot deliver the expected security value unless the programs (code) respect these security artefacts that have been produced in the preceding stages. This sets the stage for model driven security in which transformations of architecture and design artefacts is essential, as well as the verification of code compliance with various properties. Some of these properties have been embedded in the security specific elements of the software design; other may simply be high priority security requirements that have articulated – such as the appropriate treatment of concurrency control and the avoidance of race conditions in the code, as a typical FI service in the cloud may be deployed with extreme concurrency in mind.

Supporting security requirements in the programming – code – level requires a comprehensive approach. At least two essential facets must be covered:

- The service *creation* means must be improved and extended to deal with security needs. Service creation means aggregating as well as *composing* services from pre-existing building blocks (services and more traditional components), as well as *programming* new services from scratch using a state-of-the-art programming language. The service creation context will typically aim for techniques and technologies that support compile and build-time feedback. One could argue that security support for service creation must focus on and enable better static verification.
- The service *execution* support must be enhanced to deal with hooks and building blocks that facilitate effective security enforcement at run-time. Dependent on the needs and the state-of-the-art this may lead to interception and enforcement techniques that “simply” ensure that the application logic consistently interacts with underpinning security mechanisms such as authentication or audit services. Otherwise, the provisioning of the underpinning security mechanisms and services (e.g. supporting mutual non repudiation, attribute based authorization in a cloud platform etc.) will be required as well for many of the typical FI service environments.

It is crucial to improve upon these two facets in order to lift the current state of practice to a higher degree of quality. Having the right compilation tools will not only reduce the number of bugs and help find them quicker, but it will cut down on the attack surface of an application by

avoiding common programming vulnerabilities. Additionally, if an attacker succeeds in exploiting a service, the monitoring tools and policies will be able to mitigate the attack by constraining the access of the attacker to the system.

In the remainder of this section, we further elaborate on the needs and the objectives of community wide research activities, in order to deal effectively with the grand challenge sketched above.

### **5.3.1 Middleware Aspects**

The research community should re-investigate service-oriented middleware for the Future Internet, with a special emphasis on enabling deployment, access, discovery and composition of pervasive services offered by resource-constrained nodes. The most relevant Quality-of-Service aware dynamic service discovery and composition, in particular accounting for properties related to security, privacy and trust. In order to ensure that published security properties of FI services are correct, monitoring business compositions must be monitored and analysed. Monitoring infrastructures for several platforms including Java and BPEL must be developed. Another important facet in this respect is information flow analysis for business process languages. The increasing usage of IT systems in practical business logic execution entails the need for high quality and reliability. Business workflows frequently act on behalf of multiple parties having potentially differing interests, thus malfunction can lead to the compromise of sensitive business data. Therefore, the analysis whether the business process conforms to corporate information security policies is of high priority. Note that contemporary languages and technologies lack this capability.

### **5.3.2 Secure Service Programming**

Many security vulnerabilities arise from programming errors that allow an exploit. Future Internet will further reinforce the prominence of highly distributed and concurrent applications, making it important to develop methodologies that ensure that no security hole arises from implementations that exploit the computational infrastructure allowed by Future Internet. The research community must further investigate advances over state-of-the-art in fine-grained concurrency to enable highly concurrent services of the Future Internet, and will improve analysis and verification techniques to verify, among others, adherence to programming principles and best practices.

#### **5.3.2.1 Verifiable Concurrency**

Lock-free wait-free algorithms for common software abstractions (queues, bags, etc) are one of the most effective approaches to exploit multi-core parallelism. These algorithms are hard to design and prove correct, error-prone to program, and challenging to debug. Their correctness

is crucial to the correct behaviour of client programs. Research should now focus on build independently checkable proofs of the absence of common errors, including deadlock, race conditions, and non-serialize-ability [JPS+08].

### **5.3.2.2 Adherence to Programming Principles and Best Practices**

Programming support must include methods to ensure the adherence of a particular program to well-known programming principles or best practices in secure software development. Emphasis will be put on language extensions that guarantee adherence to best practices, and verified design patterns that can be used during development. Also, it is necessary to consider that new features in programming languages may result in new features in modelling languages. Moreover, new programming constructs will arise to deal with several security properties and include disciplined programming techniques.

The research community might investigate and re-visit methods from language-based security, in particular type systems, to enforce best-practises currently used in order to prevent cross-site scripting attacks and similar vulnerabilities associated with web-based distributed applications. Obviously, the logical rationales underlying such best practises must be articulated, enabling the development of type systems enforcing these practises directly – thus allowing users to deviate from rigid best practices while still maintaining security.

### **5.3.3 Platform Support for Security Enforcement**

Future Internet applications span multiple trust domains, and the hybrid aggregation of content and functionality from different trust domains requires complex cross-domain security policies to be enforced, such as end-to-end information flow, cross-domain interactions and usage control. In effect, the security enforcement techniques that are triggered by built-in security services and by realistic in the FI setting, must address the challenge of *complex interactions* and of *finely grained control* [HMS06]. Research should therefore focus on the enforcing cross-domain barriers in the interaction among different cross-domains, and on the enforcement of fine-grained security policies via execution monitoring.

#### ***Secure Cross-Domain Interactions***

Web technology inherently embeds the concept of cross-domain references, and applications are isolated via the Same-Origin-Policy (SOP) in the browser. From a functional perspective, the SOP puts limitations on compose-ability and cooperation of different applications, and from a security perspective, the SOP is not strong enough to achieve the appropriate application isolation.

#### ***Finely grained execution monitoring***

Trustworthy applications need run-time execution monitors that can provably enforce advanced security policies [GBJ06][BLW05] including fined-grained access control policies usage control policies and information flow policies [SM03]. (This topics is clearly related also to the area of run time verification and enforcement.)

### ***Supporting Security Assurance for FI Services***

Assurance will play a central role in the development of software-based services to provide confidence about the desired security level. Assurance must be treated in a holistic manner as an integral constituent of the development process, seamlessly informing and giving feedback at each stage of the software life cycle by checking that the related models and artefacts satisfy their functional and security requirements and constraints. Obviously the security support in programming environments that must be delivered will be essential to incept a transverse methodology that enables to manage assurance throughout the software and service development life cycle (SDLC).

#### **5.3.4 Threats**

OWASP project ([www.owasp.org](http://www.owasp.org)) top ten list for web application security clearly shows how coding issues as injection, cross scripting and generally speaking wrong programming practices are the major issues to be tackled. Indeed, reliable programming environments and proper coding techniques are crucial to minimize the presence of exploitable vulnerabilities in software-based services. Lack of specific research activities in the previous topics risks to contribute to have errors in those programming environments that will be also propagated to the final programmer code. Similarly, failing at recognizing that new high level service execution languages introduce new potential threats (in addition to the level of system programs) as well as the need of middleware for run-time monitoring risks to contribute to expand the possibility of attacks on web services.

### **5.4 Secure Service Composition and Adaptation**

Future Internet services and applications will be composed of several services (created and hosted by various organizations and providers), each with its own security characteristics. The business compositions are very dynamic in nature, and span multiple trust domains, resulting in a fragmentation of ownership of both services and content, and a complexity of implicit and explicit relations among the participants. Service composition support is required, in terms of the composition languages a la BPEL, as well is in terms of the underpinning in middleware platforms.

One of the challenges for the secure service composition is the need for new formalisms to specify service requests (properties of service compositions) and service capabilities, including their security policies, and tools to generate code for service compositions that are able to fulfil these requirements based on the available services. In addition to complying with the requested functional and quality-of-service-related characteristics, composition languages must support means to preserve at least the security policy of those services being composed.

As a matter of fact, dynamic adaptation will play a major role in FI applications to ease service composition, paying special attention to the semantic level adaptation, which is left aside in most of the nowadays proposals. Security contracts should be used during the whole life of software and will be exposed in the composition of services, not just in single services, and their dynamic evolution should be managed. Furthermore, the existence of an open market for composable services with well-defined security properties is required, and service customization should not come at the cost of security. As a consequence of the reasons mentioned above, service composition should be an easy, secure, and commonly performed task.

Given that the outcome of the composition of two secure services might not be a secure bigger service, it is required to assess the risk of a service composition. Also, it would be very interesting to have a test-bed for comparing Service Adaptation by Contract approaches. Other topics to address include quantifying and control information sharing in service composition, and developing automatic risk reduction capabilities when recruiting services for compositions. Unifying the different Aspect-Oriented Modelling (AOM) techniques for model composition poses a gap. Also, the research community needs to consider the cases where only partial or inadequate information on the services is available, in such a way that the composition will have to find compliant candidates or uncover the underspecified functionality

In order to achieve the integration and interoperability of services, some on-going solutions are based on semantic annotations and secure adaptation contracts, as well as on decentralized secure composition and on distributed component models. However, further solutions are required. First, services and components need to be more open, with clearer open interfaces and need to be easily accessible from known repositories. Moreover, it is required to research on how to efficiently compose security measures.

#### **5.4.1 Threats**

Service composition is one of the main distinguished features of Future Internet service paradigm. The capability to achieve trustworthy secure composition is thus paramount. Failing in constructing such a framework would harm the whole concept. Building secure services, that cannot be further composed is an inherent obstacle that need to be removed and would make fragile the whole architecture.

### **5.5 Run time verification and enforcement**

An important set of activities relates to **run-time verification**. Run-time *verification* complements programming-level verification and testing in order to provide the assurance that the latter cannot always deliver, be it for scientific and technological reasons, be it for reasons of organizational complexity. The latter may frequently occur in a multi-organisational context, typical for service compositions in Future Internet. We need to study approaches for run-time monitoring of data flow, as well as technologies for privacy-preserving usage control.

- *Run-time monitoring of data flow.* Electronically and autonomously executed business logic plays a crucial role in today's practice. Since these systems may possibly have access to sensitive data of different parties with potentially contradicting interests, information flow policies may need to be enforced. Lately it has been shown that information flow controlling run-time monitors can assure the same level of termination insensitive non-interference as the original Denning style static checking procedure, while providing the advantage of being able to be more permissive. We need to develop the theoretical foundations of a run-time monitor, which is suited for the enforcement of information flow policies in an environment, where complex hierarchic data is manipulated (such as for instance in BPEL).
- *Monitoring Usage Control Properties.* Usage control (e.g. see [LMM10]) extends traditional access control with policies and mechanisms to control the usage of data after it has been accessed. Therefore, usage control addresses central privacy-related security issues, which are raised by Future Internet applications and for which only partial solutions exist nowadays. The objective of this activity is to advance the state of the art in observing and controlling the usage of sensitive data in Future Internet applications. There is the need to develop methods that monitor the use of data and ensure that usage conforms to the intended purposes for which the data was collected. Based on previous work on usage control and monitoring of security policies, we need to adapt and extend run-time verification techniques for checking the adherence of data consumers to usage control policies. Furthermore, we will study the integration of these monitors into Future Internet applications that report on or, where possible, prevent the misuse of sensitive data.

A current limitation in monitoring security policies is the amount of data that can be efficiently processed with the existing monitoring techniques. We need to consider specific techniques addressing this issue. Moreover, security policies are typically formulated at high levels of abstraction whereas the monitors observe low-level system events that are scattered to different network nodes or to different layers of the system stack. The management of distributed enforcing mechanisms also deserves further study.

### **5.5.1 Threats**

Run-time verification and enforcement offer the possibility to overcome some limitations of other security techniques as static analysis and permits continuous and fine-grain enforcement of policies. Lack of these technologies will limit the control capabilities that user may exercise on systems/services/data, as well as harm the possibility to enable compliance techniques with legislations, organizational policies, and business rules. As a matter of fact, run-time verification and enforcement are recognized as very valuable tools.

## **5.6 Users Security Awareness**

The growth of the FI is making users become more situational-security aware, that is, users are increasing the knowledge on the threats and risks they are exposed.. However, security is a moving target and security requirements are possibly the most dynamic type of requirements. Vulnerabilities and threats are published on a daily basis and it is difficult to keep the pace even for the experts in dedicated ICT companies. These dynamicity poses several issues such as for example how to present clear information to the user about changes in security concerns. Usability, privacy controls and security configuration for end-users should be context-aware. In regard to privacy, similar to other “high-level” declarative requirements that link natural language to operational level events, a key research direction is the link between machine-readable and composable declarative policies that make easier policy monitoring or enforcement and natural language policies that facilitate understanding by end-users and compliance with regulations. In this area, the semantic technologies should be adopted by wide range of security mechanisms, policies and solutions, in addition to a visual representation of security state for awareness and analysis.

### **5.6.1 Threats**

If we lose the opportunity to perform research in this direction, users would consider security as a “show stopper” and this would allow the spread either of services with limited security mechanisms, thus attack-prone, or disincentive user from using those services.

## **5.7 Security management**

The horizontal activities in the area of full life cycle support for secure services are predominantly addressing the construction, and to a lesser extent the potential evolution and adaptation of secure services for the future internet. It is clear that such secure services – especially the security features and related subsystems - should be supported with appropriate monitoring and management support in order to observe the “quality of protection” in production systems at run time, and in order to implement the necessary measures for dealing with new threats and attacks, and possibly also with security incidents that require modification of the service implementation and/or its deployment environment. This type of activity (security monitoring and management) is not new in the domain of monitoring security infrastructures and secure systems, yet it is still limited in the space of service provisioning and deployment.

It should be noticed that the research on risk management and assurance, as sketched in earlier sections, will be instrumental to this additional challenge. Security monitoring and management for new services will be essential to limit and control the total cost of ownership in the corresponding services business. This topic should also be considered in light of the

“autonomic security” challenge, which may offer additional advantages that contribute to the business case of securing FI services.

### **5.7.1 Threats**

Security management is always a main concern. The new FI services demand for new management techniques to ease system and services administrator tasks, in particular during security incidents, allowing them to manage and maintain secure systems.

## **5.8 Autonomic Security**

In the future, due to a high number of events, devices , users, services etc, automated customization at run-time of specific security mechanisms will be of paramount importance, although it could lead to new potential attacks in which the attacker tries to blow down this automated reconfiguration mechanism.

Autonomic security assumes security decisions as autonomous and spontaneous act of the system, considering the possibility to take appropriate actions, based on self-capabilities, as self-monitoring and self-protection.

In this ambit, predictive analysis of security problems that can be used in order to anticipate and rely on a good decision support is challenging. The key issues for this are twofold, creating a smart reasoner that makes rapid and relevant reconfiguration decisions, and deciding which data is really valuable for feeding the reasoner.

Secure dynamic adaptive architectures is another challenging field, more specifically how to integrate security concerns in these adaptive architectures, and how to consider specific features of FI (e.g., monitoring geo-localization to adapt security according to the location of a mobile device). Understanding the effect of architecture reconfiguration on the application is a gap to be filled, being required verification based on concern interaction analysis.

Solutions should focus on new suitable monitoring mechanisms to feed the reasoners. In order to tackle evolution of FI environments, adaptive configuration of policies and countermeasures, as well as dynamicity of mechanisms to respond to vulnerabilities, are required. Enforcement gateways with reacting components could take a more relevant role as well. Establishment of security contexts under which a service executes are also relevant, since reconfiguration decisions will be based on these contexts. The research directions in this area are twofold: development of contextual frameworks for security, and verified reusable components for certain contexts.

### **5.8.1 Threats**

The main risk of not having autonomic security is to fall again in the well-known problem of having security detached from system development. In that case, security is just considered as an afterthought, with all the costs deriving from system re-design/development and deployment. As a matter of fact, considering autonomic security helps to consider security as a built-in aspects of any ICT product.

## 6 Conclusions

This document briefly illustrates various lines of research that we consider useful in the mid-term in the area of engineering secure software-based services. This document expands and revises the white paper published in [JLMM11]. Clearly, the needs and challenges sketched in this document reach beyond the scope and capacity of a closed consortium and needs to be addressed by a larger community. The topics listed above should be shared and tackled by an entire and open research community.

## References

- [BCK04] Len Bass, Paul Clements, and Rick Kazman, *Software Architecture In Practice*, Addison-Wesley, 2004
- [BGG+04] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. *TROPOS: An Agent-Oriented Software Development Methodology*. In *Journal of Autonomous Agents and Multi-Agent Systems*, 8(3):203--236, 2004.
- [BHS+07] F. Braber, I. Hogganvik, M. S. Lund, K. Stølen, F. Vraalsen Model-based security analysis in seven steps – a guided tour to the CORAS method. In *BT Technology Journal*, pages 101-117, Springer, 2007
- [CSB+08] Manuel Clavel, Viviane da Silva, Christiano Braga, and Marina Egea, *Model-Driven Security in Practice: An Industrial Experience*. In *Proceedings of ECMDA-FA'08 (4<sup>th</sup> European Conference on Model Driven Architecture: Foundations and Applications)*: 326-337, Berlin, Germany, 2008
- [DLF93] A. Dardenne, A. van Lamsweerde, and S. Fickas. *Goal-directed Requirements Acquisition*, *Science of Computer Programming*, 20(1--2): 3--50, 1993.
- [EFP11] Effectsplus Cluster Meeting, Trust and Security Roadmapping Report, <http://www.effectsplus.eu/files/2011/04/Trust-and-Security-Research-Roadmap-Draft-1.pdf>, 29-30 March, 2011
- [BLW05] L. Bauer, J. Ligatti, and D. Walker. Composing security policies with polymer. In *PLDI*, pages 305–314, 2005.
- [HMS06] Kevin W. Hamlen, Greg Morrisett, and Fred B. Schneider. Computability classes for enforcement mechanisms. *ACM Trans. Program. Lang. Syst.*, 28(1):175–205, 2006.
- [SM03] Andrei Sabelfeld and Andrew Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 2003.
- [GBJ06+] G. Le Guernic, A. Banerjee, T. P. Jensen, D. A. Schmidt: Automata-Based Confidentiality Monitoring. *ASIAN 2006*: 75-89
- [JFS+08] B. Jacobs, F. Piessens, J. Smans, K. R. M. Leino, W. Schulte: A programming model for concurrent object-oriented programs. *ACM Trans. Program. Lang. Syst.* 31(1): (2008)
- [JLMM11] [Wouter Joosen](#), [Javier Lopez](#), Fabio Martinelli, [Fabio Massacci](#): Engineering Secure Future Internet Services. [Future Internet Assembly 2011](#): 177-192

[FFR+07] R. France, F. Fleurey, R. Reddy, B. Baudry, and S. Ghosh. *Providing Support for Model Composition in Metamodels*. In Proceedings of EDOC'07 (Enterprise Distributed Object Computing Conference). Annapolis, MD, USA, October 2007.

[GBS06] S. Gürses, B. Berendt, and T. Santen. *Multilateral security requirements analysis for preserving privacy in ubiquitous environments*, In Proc. Workshop on Ubiquitous Knowledge Discovery for users (UKDU'06), 2006.

[GMZ06] P. Giorgini, H. Mouratidis, and N. Zannone. *Modelling Security and Trust with Secure Tropos*. In: Integrating Security and Software Engineering: Advances and Future Vision. 2006. IDEA. ISBN 1-59904-149-9

[ISO09] International Organization for Standardization. ISO 31000 Risk management - Principles and guidelines, 2009.

[KWF08] A. Kubo, H. Washizaki, and Y. Fukazawa, *Extracting Relations among Security Patterns*, In Proceedings of SPAQu'08 (International Workshop on Software Patterns and Quality), 2008

[LMM10] [Aliaksandr Lazouski](#), Fabio Martinelli, Paolo Mori: Usage control in computer security: A survey. Computer Science Review 4(2): 81-99 (2010)

[MFB+08] Brice Morin, Franck Fleurey, Nelly Bencomo, Jean-Marc Jézéquel, Arnor Solberg, Vegard Dehlen, and Gordon Blair, *An aspect-oriented and model-driven approach for managing dynamic variability*. In MoDELS'08, Toulouse, France, October 2008

[OG] Open Group, Security Design Pattern Technical Guide, <http://www.opengroup.org/security/gsp.htm>

[RFL10] David G. Rosado, Eduardo Fernández Medina, and Javier López, *Security services architecture for Secure Mobile Grid Systems*, Journal of Systems Architectures, 2010

[WMA+07] Jon Whittle, Ana Moreira, João Araújo, Praveen K. Jayaraman, Ahmed M. Elkhodary, Rasheed Rabbi: *An Expressive Aspect Composition Language for UML State Diagrams*. MoDELS 2007: 514-528, 2007

[YWM08] N. Yoshioka, H. Washizaki, and K. Maruyama, *A survey on security patterns*, Progress in Informatics, N.5, pp.35-47, 2008

## Appendix A: Roadmap Questionnaire

<b>Partner Name</b>	
<b>Research Areas (regarding the map)</b>	

	Short-mid term (2-4 years)	Long term (5-10 years)
<b>Overall Vision</b>		
Are there any missing or overlapping topics in the map? Are there wrong or missing links amongst topics?		
How may your area of expertise change or influence on the current SDLC?		
How may your area of expertise influence on the progress of other research topics included in the map?		
How do you think other research areas may influence on the progress of your own?		
Describe your level of cooperation between academia and industry		

<b>Ongoing Changes</b>		
What are the current gaps or unsolved problems in your research area?		
What do you think it is changing in your area within the FI paradigm?		
<b>New Challenges and Gaps to be solved</b>		
Challenges of your research topic w.r.t. the FI paradigm		
How are you proceeding in order to fill the gaps and meet the challenges?		
<b>Solutions</b>		
What solutions or approaches will you foresee to solve the challenges and fill the gaps?		
What do you think it will be the relationship between industry and academia in order to realise the achieved solutions?		
What new technologies do you foresee derived from the new research solutions or approaches? And threats?		